UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

RE-IDENTIFICACIÓN DE PERSONAS

Daniel Sáez García
Tutor: Álvaro García Martín
Ponente: José María Martínez Sánchez
Julio 2019

Re-identificación de personas

AUTOR: Daniel Sáez García
TUTOR: Álvaro García Martín
PONENTE: José María Martínez Sánchez

Video Processing and Understanding Lab
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2019

Trabajo parcialmente financiado por el Gobierno de España bajo el proyecto
TEC2017-88169-R (MobiNetVideo)

Resumen

En la actualidad, la re-identificación de personas es un recurso con una alta demanda sobre todo en el ámbito de la video seguridad, esto no significa conocer la identidad de una persona sino poder hacer un seguimiento de esta en distintas cámaras cuyas imágenes no se solapen. Hay una gran cantidad de medidas de evaluación tradicionales que nos permiten hacer extracciones" manuales" de características, las cuales utilizan algoritmos matemáticos los cuales permiten extraer información de las imágenes. Sin embargo, hay gran cantidad de aspectos a tener en cuenta si queremos que nuestros modelos funcionen lo mejor posible, como puede ser la orientación de la persona, el entorno o su posición. La extracción de características basada en aprendizaje profundo realiza un modelado de datos, para ello utiliza flujos de datos de gran tamaño para aprender de estos y poder realizar una clasificación y un análisis predictivo. El aprendizaje profundo se basa en la utilización de redes neuronales, un modelo matemático que trata de imitar el comportamiento biológico de las neuronas, conectando diferentes capas de procesamiento y otorgando distintos pesos a cada una con el fin de obtener un modelo optimizado.

El objetivo de este TFG es la comparación de los resultados de las medidas de extracción manuales (handcrafted features) y las automáticas (basadas en Deep Learning), esto se realizará ejecutando un script que nos calcule el porcentaje de acierto a la hora de re-identificar personas entre las cámaras utilizando los distintos métodos de extracción manual y después utilizando los basados en redes neuronales en los datasets que utilicemos para evaluar.

Palabras clave

Aprendizaje profundo, redes convolucionales, características tradicionales, identificación, extracción de características, métrica de aprendizaje.

Abstract

Nowadays, person re-identification is a resource with a high demand especially in the field of video surveillance, this does not mean knowing the identity of a person but being able to monitor it in different cameras whose images are not overlapped. There are a lot of traditional evaluation measures that allow us to make "manual" feature extractions, which use mathematical algorithms which allow extracting information from the images. However, there are many aspects to consider if we want our models to work as well as possible, such as the orientation of the person, the environment or their position. Feature extraction based on deep learning makes a modeling of data, for this task, uses large data flows to learn from these and to perform a classification and predictive analysis. Deep learning is based on the use of neural networks, a mathematical model that tries to imitate the biological behavior of the neurons, connecting different layers of processing and granting different weights to each in order to obtain an optimized model.

The task of this bachelor thesis is the comparison of the results of the manual extraction measures (handcrafted features) and the automatic ones (based on Deep Learning), this will be done by executing a script that calculates the percentage of success when it comes to identify people between the cameras using the different methods of manual extraction and then using those based on neural networks in the datasets we use to evaluate.

Keywords

Deep learning, Convolutional network, Handcrafted features, Identification, Feature extraction, Metric learning.

Agradecimientos

Quiero darle las gracias a mi tutor Álvaro por la ayuda que me ha brindado durante la realización del trabajo.
También agradecer a mi familia y amigos por su confianza y todo el apoyo que han sido para mí. Gracias a todos los que he conocido todos estos años, que el camino se ha hecho juntos.

Muchas gracias a todos.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Motivación, Objetivos y Organización

1.1 Motivación

La re-identificación de personas es un asunto que está a la orden del día, especialmente en el ámbito de la video seguridad, esta motivación afecta tanto al sector público como al privado. El auge de esta tecnología es debido en gran parte al gran incremento de cámaras de seguridad tanto en las calles como en los edificios, ya sean privados o públicos.

Hasta la fecha, para desarrollar esta tarea, se ha realizado mediante medidas tradicionales o handcrafted features, ésta realiza cálculos matemáticos para identificar a las personas, algunas de estas medidas son muy efectivas.

Además, las técnicas de aprendizaje automático o *Machine Learning*, en especial el aprendizaje profundo o *Deep Learning* y la publicación de grandes datasets con imágenes extraídas de cámaras de seguridad ha ayudado a la posibilidad de implementar modelos de identificación de personas mediante el uso de redes neuronales convolucionales (CNN por sus siglas en inglés). El funcionamiento se basa en imitar el comportamiento de las neuronas, es decir, se creará una red compleja formada por numerosas capas conectadas entre sí en las que se realizarán una serie de cálculos y se compartirán con las capas posteriores. Después se generará una salida con datos de gran relevancia para que el ordenador aprenda de manera automática.

Por ello el objetivo queremos lograr con este TFG es realizar una comparación entre las medidas basadas en aprendizaje profundo con el fin de ver la gran utilidad de las redes neuronales y las medidas tradicionales, viendo cuál o cuáles son más efectivas a la hora de realizar la re-identificación de personas. Esto se llevará a cabo entrenando distintas redes con distintos dataset para poder evaluar luego sobre estos y las medidas tradicionales las tasas medias de acierto o ajuste a la re-identificación sobre los datasets.

1.2 Objetivos

En el ámbito de la video vigilancia, la re-identificación de personas juega un papel fundamental, esta no necesita conocer la identidad exacta de un usuario sino poder determinar que se trata del mismo individuo al ser captado en distintas cámaras, pudiendo realizar la identificación en función de las características de cada individuo, su postura, vestimenta, físico, etc. La capacidad actual de tratar grandes flujos de datos creando modelos utilizando redes neuronales supone un paso hacia delante en esta tarea, ahorrando tiempo en analizar imágenes con respecto a las medidas tradicionales.
El objetivo de este TFG consiste en realizar una comparativa de los resultados obtenidos al utilizar varias redes neuronales entrenadas previamente con distintos en entornos de prueba, datasets, y evaluar el porcentaje de acierto al utilizar estas redes y de las medidas de extracción de características manuales para ver cuáles son más efectivas.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1:** Motivación, objetivos y organización de la memoria.
- **Capítulo 2:** Estado del arte.
- **Capítulo 3:** Diseño y desarrollo
- **Capítulo 4:** Integración
- **Capítulo 5:** Pruebas y resultados
- **Capítulo 6:** Conclusiones y trabajo futuro
- **Bibliografía**
- **Glosario**

2 Estado del arte

2.1 Introducción

Durante este trabajo de fin de grado se trata como tema principal la re-identificación de personas, tomando como referencia el estado del arte de los artículos [1][2][3][12][13]

En los circuitos cerrados de cámaras se obtiene una gran cantidad de información, poder trabajar con estas como un conjunto y no de manera independiente es de gran utilidad. Esto presenta complicaciones ya que en una imagen el sujeto a re-identificar puede haber cambiado su postura, puede ser captado desde otro punto de vista, aparezca con mayor o menor tamaño, que las condiciones de iluminación sean distintas, etcétera. Otra complicación importante es que al no estar aislados los individuos se producen situaciones que dificultan el análisis como que se tapen partes del cuerpo en algunas situaciones.

El autor del artículo [18] propone una evaluación metódica utilizando la tecnología basada en el estado del arte en cuanto a extracción de características, métricas de aprendizaje y evaluación de resultados.

En las siguientes páginas veremos el estado del arte de los artículos anteriormente citados, divididos en varios por su temática, en estos se habla del funcionamiento de un sistema de re-identificación, el uso de datasets y la utilización de redes neuronales convolucionales para la re-identificación.

En el ámbito general de la re-identificación, el autor del artículo [14] expone que el estado del arte divide en dos tipos de técnicas de re-identificación:

- *Single-Shot*: Este método tratara de asignar o relacionar pares de fotos que contienen capturas de un individuo por secuencia. Se registrarán sus variaciones del color en una matriz, esto funciona bien si las variaciones en los puntos de vista no son muy grandes.

- *Multi-Shot:* En este método hay varias imágenes de cada individuo en la secuencia de vídeo de la cámara. Aquí se busca aumentar el poder discriminativo de cada individuo respecto del resto, se entrenará un modelo con varias imágenes de cada persona.

2.2 Estructura de un sistema de re-identificación

Un sistema de re-identificación presenta la siguiente estructura:

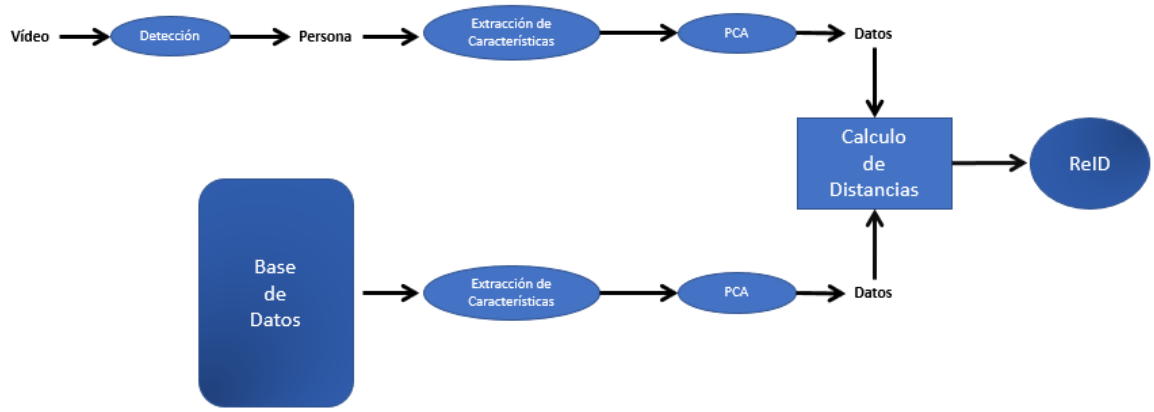

Figura 1. Esquema de un sistema de ReID

El esquema de la figura 1 muestra paso a paso el proceso llevado a cabo, este TFG se encarga de toda la parte posterior a la detección, buscará coincidencias con la base de datos y las nuevas personas, se establecerá un modelo de evaluación y un posterior reajuste con la finalidad de ir mejorando este modelo a medida que trabaje la información y a la salida se ve que están las personas ya identificadas.

En la fase de captación el autor del artículo [12] manifiesta que el estado del arte es la utilización varias cámaras de vigilancia para la re-identificación de personas.

En la etapa de detección, el estado del arte, explicado por el autor en el artículo [1] habla acerca de una gran mejora de resultados al combinar las distintas imágenes de una persona utilizando homografías y correlación entre imágenes. Por otro lado, el autor del artículo expone en el estado del arte el uso de modelos de partes deformables (DPM por sus siglas en inglés) el cual consiste en detectar por una parte a la persona al completo y por otra realizar una segmentación de las partes del objeto a análisis (en nuestro caso brazos cabeza torso piernas, etc).

En el artículo [3] el autor nos explica el estado del arte de los dataset y la incorporación de datos adicionales generados mejorando así los resultados obtenidos en el campo de la re-identificación de personas.

Para la etapa de extracción de características el estado del arte se basa en el aprendizaje automático y en concreto en el aprendizaje profundo, el autor del artículo [2] propone analizar las imágenes segmentándolas e identificando las distintas partes del cuerpo para poder mejorar en aquellas imágenes en las que los individuos están parcialmente tapados.

2.2.1 Captación

Dependiendo de las características de nuestro entorno de grabación (al aire libre, espacios cerrados, tamaño, iluminación, etc) habrá que realizar un diseño u otro de la red de cámaras. Hay que distinguir también entre si la red tiene habilidad de captar una o varias imágenes por persona al mismo tiempo esto determinará si el sistema es *single-shot* o *multi-shot.* La ventaja de un sistema *single-shot* reside en su simplicidad, sin embargo, un sistema *multi-shot* será más robusto ya que podrá extraer más información al tener más puntos de vista, por ejemplo, una única captura solo tendrá un punto de vista, el cual puede ser bueno o malo, un sistema multi-cámara reducirá este problema al haber más imágenes y por tanto distintos puntos de vista de la persona.

2.2.2 Detección

Una vez tenemos el vídeo de la cámara, este entra en la etapa de detección, esta es la encargada de analizar todos los *frames* de la secuencia de vídeo para después determinar las zonas de la imagen en las que hay personas, segmentación de personas. Aquí hay múltiples factores a tener en cuenta, para realizar esta detección se realizará una de la misma manera que en la búsqueda de objetos en primer plano, asumiendo que lo único que se mueven son las personas.

2.2.3 Extracción de características

Después de haber detectado a las personas en el entorno de vigilancia, se procede a la extracción de características, estas se dividen en tradicionales (*hancrafted*) y obtenidas mediante aprendizaje automático (*non-handcrafted*), en nuestro caso utilizaremos ambas.
Para el caso de las tradicionales, utilizaremos los métodos WHOS (*Weighted Histogram of Overlapping Stripes*), GBICOV (*Biolgical inspired features combined with Covatiance*), LDFV (*Local Descriptors encoded by Fiher Vector*), *Color & Texture, Histogram* y LBP (*Local Binary Pattern*).
En el caso de las medidas basadas en aprendizaje automático, entrenaremos las redes neuronales Alexnet, Resnet-18 y VGG-16.

2.2.4 Algoritmo PCA

Una vez extraemos nuestros datos aplicaremos el algoritmo PCA (*Principal Component Analysis*) para reducir la dimensionalidad de nuestros datos, esto se realiza obteniendo nuevas características denominadas componentes los cuales se sacan de los datos que no están correlados entre sí. Es un algoritmo no supervisado, es decir, los datos no están etiquetados. Este se divide en 3 etapas:

1. Normalización de variables.
2. Obtención de autovalores y autovectores de la matriz de covarianzas.
3. Creación de la matriz de proyección y transformación de los datos de inicio.

2.2.5 Métricas de aprendizaje

Una vez hemos obtenido los resultados de nuestros datos y reducido su dimensionalidad, habrá que efectuar un cálculo de las distancias de cada registro. Hay varios métodos sobre los que calcular las distancias, el objetivo es que para una misma persona la distancia sea lo más pequeña posible y para otras que ocurra todo lo contrario. El autor del artículo [18] propone utilizar los métodos FDA (*Fisher Discriminant Analysis*) [26], LFDA (*Local Fisher Discriminant Analysis*) [27], KLFDA (*Kernelized Local Fisher Discriminant Analysis*) [28], MFA (*Marginal Fisher Analysis*) [29], KMFA (*Kernelized Marginal Fisher Analysis*) [28], XQDA (*Cross-view Quadratic Discriminant Analysis*) [30], PCCA (*Pairwise Constrained Component Analysis*) [31], RPCCA (*Regularized Pairwise Constrained Component Analysis*) [31], KPCCA (*Kernelized Pairwise Constrained Component Analysis*) [31], NFST (*Null Foley-Sammon Transform*) [32], KISSME (*Keep-It-Simple-and-Straightforward-Metric*) [33], PRDC (*Probabilistic Relative Distance Comparison*) [35], SVMML (*Support Vector Machine on Multi Layer*) [34].
Para separar los datos por clases se utilizan *kernels*, estos establecen unos rangos de decisión para catalogar un dato en una clase u otra, en nuestro benchmark, estarán disponibles los siguientes: Lineal, Chi cuadrado, Chi cuadrado-rbf y Exponencial.

3 Diseño y desarrollo

3.1 Introducción

En este capítulo vamos a hablar acerca de los distintos parámetros de evaluación y aprendizaje utilizados durante la realización de este trabajo de fin de grado.
Vamos a ver como se entrenan las redes neuronales y los distintos aspectos a tener en cuenta: pesos, ajuste, datos de entrenamiento, datos de test, etc. Además, se van a explicar los cambios realizados en los distintos scripts para utilizar el sistema de evaluación en redes neuronales. Por último, veremos los distintos parámetros de ejecución a la hora de extraer los resultados.
Los métodos de extracción de características, como ya hemos dicho anteriormente, se dividen en tradicionales (handcrafted) y obtenidas mediante aprendizaje automático (non handcrafted).
El objetivo de este TFG es comparar las distintas características, por lo tanto, en nuestro script de benchmark nos centraremos en cambiar los métodos de extracción de características.

3.2 Extracción de características tradicionales

Tras haber hablado del estado del arte en el capítulo anterior, ahora haremos un análisis más detallado de estas, al ser esta parte la que cubre la motivación principal del trabajo analizaremos algunos de los métodos de extracción tradicionales.

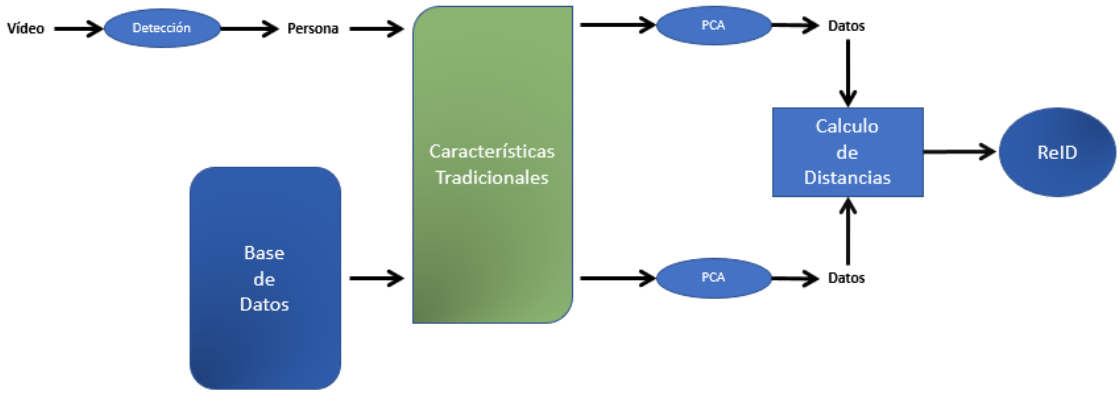

Figura 2. Esquema de re-identificación basado en características tradicionales

3.2.1 WHOS

El método de extracción de características WHOS [4] (*Weighted Histogram of Overlapping Stripes*), este realiza una superposición de las franjas consiguiendo que no haya variación causada por las condiciones de iluminación y mantiene la correlación de la información de color de las franjas vecinas.

3.2.2 gBiCov

Este extractor [3] combina tres técnicas, la primera basada en el sistema visual humano (BIF), la segunda consiste en utilizar un descriptor de la covarianza y por último una combinación de los dos primeros. Estas técnicas proporcionan robustez a cambios en la iluminación, la forma y en el fondo.

3.2.3 LDFV

Una abreviatura de *Local Descriptors encoded by Fisher Vector* [6] , este describe la imagen utilizando un vector de 7 coordenadas, las coordenadas de los pixeles en x, y, la intensidad del píxel en función de ambas coordenadas, las primeras derivadas parciales de la intensidad del píxel y las segundas derivadas parciales. Este modelo utiliza un modelado gaussiano con estimación de alta probabilidad. Tras realizar los cálculos en las imágenes se realizará un cálculo de distancia euclídea para evaluar la similitud.

3.2.4 Color & Texture

Estás características [7] son analizadas mediante la aplicación filtros de Gabor y de Schmid sobre las componentes RGB, HSV e YCbCr, permitiendo diferenciar las imágenes por diferencias en la intensidad.

3.2.5 Histograma LBP

En este método, primero se aplica el algoritmo LBP [8][9] (Local Binary Pattern) por cada canal en caso de ser a color o en uno si se trabaja en escala de grises. Se va píxel a píxel asignando un ‘0’ o un ‘1’ a sus 8 vecinos según si su componente es menor (0) o
mayor (1) que el central, con estos 8 dígitos se genera un numero binario y se realiza el histograma. Es un método útil para la textura, y la disposición geométrica además de tener un bajo coste computacional.

3.3 Extracción de características basadas en redes neuronales

En este TFG en concreto nos centraremos en un aprendizaje supervisado basado en *Deep Learning* [11], esto significa que para entrenar el modelo utilizaremos datos etiquetados, es decir, al pasarle al programa la información, esta irá separada por personas.

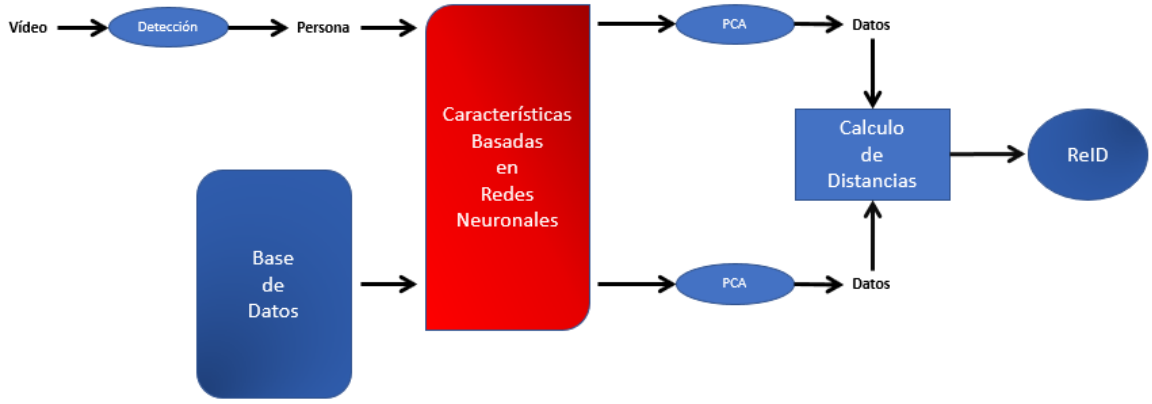

Figura 3 . Esquema de re-identificación basado en redes neuronales.

La principal ventaja de este método de aprendizaje respecto de los tradicionales es que este aprende por sí solo, solo es necesario darle información etiquetada y con eso bastará para que de manera automática comience a realizar los cálculos y a modelar los datos con gran precisión.

Las redes neuronales se dividen en tres etapas:

- Capa de entrada: Esta es la que recibe la información a modelar.
- Capas ocultas: Capas que no están en contacto con la información del exterior de la red.
- Capa de salida: Esta es la que devuelve los resultados obtenidos al como resultado de procesar la información por todo el sistema.

Una red convolucional está dividida en varias etapas, las primeras tienen dos tipos de capas, capas convolucionales (*convolutional layers*) y capas de agrupación (*pooling layers*).
Las capas de agrupación se encargan de reducir el tamaño de nuestro mapa de características, agrupando datos y devolviendo valores únicos por cada agrupación. Al reducirse el tamaño conseguiremos un aumento en la velocidad de cálculo y evitar el sobreajuste.

Las capas ocultas pueden ser de varios tipos según la función que realicen:

- Capas Convolucionales (*Convolutional Layers*): Estas se organizan por mapas de características y combinan las imágenes que entran al sistema con el banco de filtros. Para el aprendizaje se asignarán unos pesos que variarán en de la capa y del estado del entrenamiento.

- Capas ReLU (*Rectified Linear Unit*): Estas aplican a los datos de entrada, los suma y pasa el resultado por una función de activación muy simple que se encarga de poner los valores negativos a 0 y mantener los valores positivos sin modificaciones. Su gráfica tiene la siguiente forma:

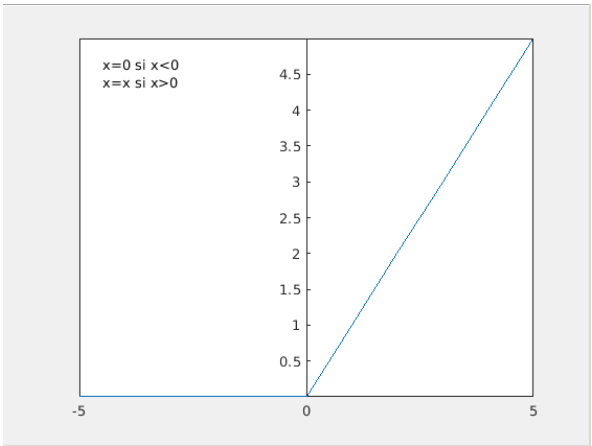

Figura 4. Función de activación ReLU

- Capa de normalización (*Normalization Batch*): Se encarga de normalizar los datos, entre 0 y 1 y así evitar que se calculen resultados incorrectos debidos distintos órdenes de magnitud.

- Capas de Agrupación (*pooling layers*): Estas agruparán regiones de datos para extraer de ellas un único valor, de esta manera el proceso se agilizará al haber menos datos y además evitar el sobreajuste.

- Capas de Descarte (*drop layers*): Estas se encargan de activar o desactivar neuronas de manera aleatoria en cada ciclo con el fin de agilizar el proceso, evitar el sobreajuste y que el aprendizaje sea lo más genérico posible.

- Capa totalmente conectada (fully connected layer): Una vez se aprenden las características hay que realizar una clasificación, esta capa genera un vector n-dimensional con las n clases que conoce nuestra red tras haber sido entrenada.

La siguiente figura [25] muestra un ejemplo de la arquitectura de una red convolucional estándar.

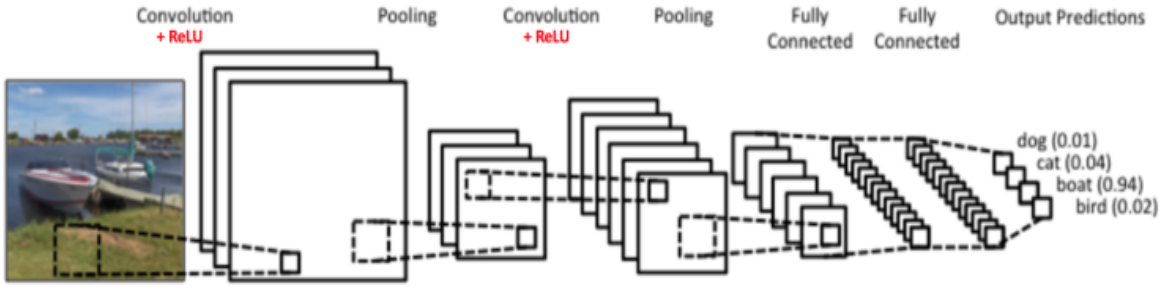

Figura 5. Arquitectura de una red neuronal estandar

En función de cómo se dispongan las capas y las conexiones entre ellas tendremos un tipo de red u otra, nosotros trabajaremos con redes estándar, con capas de adición y con capas de concatenación. Podemos ver las diferencias en la **Figura 6.**

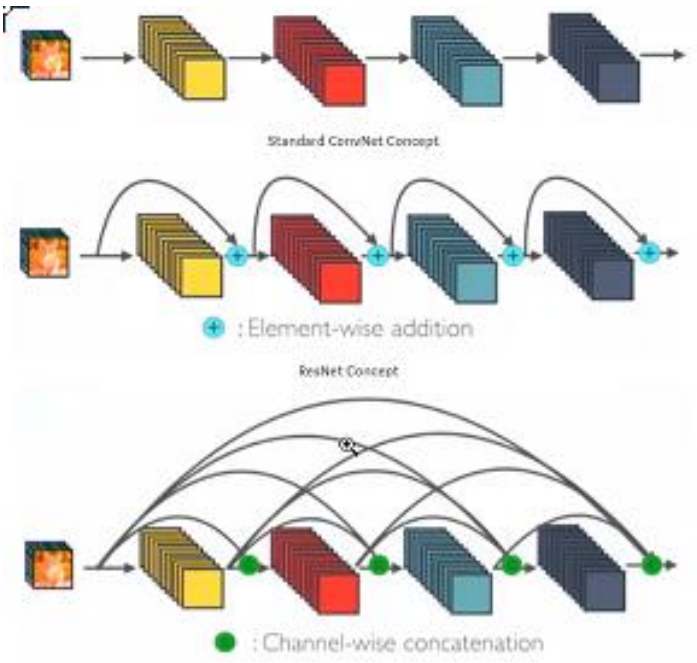

Figura 6. Ejemplo de los 3 tipos de arquitecturas

3.4 Redes neuronales utilizadas

Una vez vista la diferencia en el esquema entre el uso de técnicas tradicionales y las basadas en redes neuronales, veremos los métodos utilizados en este TFG y como se han introducido en el código para su ejecución y extracción de resultados.

En las técnicas basadas en redes neuronales, hemos utilizado 5 redes pre-entrenadas para extraer las características:

- Alexnet
- Resnet-101
- Resnet-18
- Densenet-201
- VGG-16

Estas redes están a libre disposición de los usuarios de MATLAB en su repositorio de redes pre-entrenadas. A continuación, veremos cuál es la arquitectura de estas.

3.4.1 Alexnet

Esta es la red más simple que vamos a utilizar, está formada por 25 capas de las cuales 8 tienen asignadas pesos y bias, 5 capas convolucionales y 3 fully-connected. Esta red está pre-entrenada con 1000 clases distintas y fue una de las primeras en introducir las capas de descarte para evitar el sobreajuste. Las imágenes a la entrada deben tener unas dimensiones de 227x227. Una de sus principales ventajas el que as ser una red poco profunda tiene un tiempo de procesamiento bajo. En la **Figura 7. [22]** podemos ver cuál es la arquitectura de esta red.

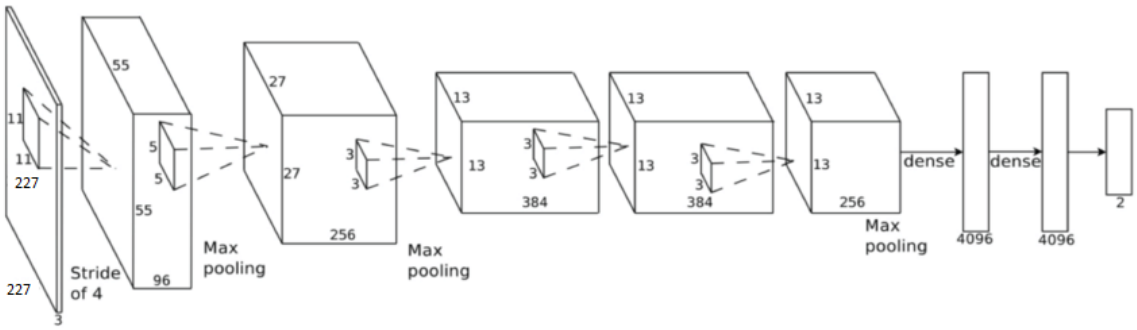

Figura 7. Arquitectura de la red Alexnet

3.4.2 Resnet

Las arquitecturas de Resnet [20] son todas ellas muy similares, solo cambiará el número de capas intermedias que las conforman, la **Figura 8 [19]** muestra una tabla con diferencias entre las redes Resnet existentes. El número que contiene la red hace referencia al número de capas que contienen pesos y por tanto parámetros que pueden ser aprendidos. La diferencia de Resnet con una red convolucional estándar es que tiene unas capas de adición en las que combina la información de capas anteriores.
Para el caso de Resnet-18, esta cuenta con un total de 72 capas con 79 conexiones entre ellas. En las 72 capas hay 18 capas denominadas *'Learnables'* las cuales tienen pesos y parámetros a aprender, la capa Conv-1 será la primera, en Conv-2 habrá otras 4 al igual que en Conv-3, Conv-4, Conv-5 y por último en la Fully Conected Layer (fc) estará la última sumando un total de 18.
Para Resnet-101 será parecido, está formada por 347 capas teniendo 101 con parámetros, en Conv-1 la primera, en Conv-2 habrá 9, Conv-3 tendrá 12, Conv-4 18 capas, Conv-5 otras 9 y la *fc* una más, alcanzando las 101.

ResNet (2015)

Layer	Output	18-Layer	34-Layer	50-Layer	101-Layer	152-Layer
Conv-1	112x112	7x7/2-64				
Conv-2	56x56	3x3 Maxpooling/2				
		$2\times\begin{pmatrix}3\times3,64\\3\times3,64\end{pmatrix}$	$3\times\begin{pmatrix}3\times3,64\\3\times3,64\end{pmatrix}$	$3\times\begin{pmatrix}1\times1,64\\3\times3\times64\\1\times1\times256\end{pmatrix}$	$3\times\begin{pmatrix}1\times1,64\\3\times3\times64\\1\times1\times256\end{pmatrix}$	$3\times\begin{pmatrix}1\times1,64\\3\times3\times64\\1\times1\times256\end{pmatrix}$
Conv-3	28x28	$2\times\begin{pmatrix}3\times3,128\\3\times3,128\end{pmatrix}$	$4\times\begin{pmatrix}3\times3,128\\3\times3,128\end{pmatrix}$	$4\times\begin{pmatrix}1\times1,128\\3\times3\times128\\1\times1\times512\end{pmatrix}$	$4\times\begin{pmatrix}1\times1,128\\3\times3\times128\\1\times1\times512\end{pmatrix}$	$8\times\begin{pmatrix}1\times1,128\\3\times3\times128\\1\times1\times512\end{pmatrix}$
Conv-4	14x14	$2\times\begin{pmatrix}3\times3,256\\3\times3,256\end{pmatrix}$	$6\times\begin{pmatrix}3\times3,256\\3\times3,256\end{pmatrix}$	$6\times\begin{pmatrix}1\times1,256\\3\times3\times256\\1\times1\times1024\end{pmatrix}$	$23\times\begin{pmatrix}1\times1,256\\3\times3\times256\\1\times1\times1024\end{pmatrix}$	$36\times\begin{pmatrix}1\times1,256\\3\times3\times256\\1\times1\times1024\end{pmatrix}$
Conv-5	7x7	$2\times\begin{pmatrix}3\times3,512\\3\times3,512\end{pmatrix}$	$3\times\begin{pmatrix}3\times3,512\\3\times3,512\end{pmatrix}$	$3\times\begin{pmatrix}1\times1,512\\3\times3\times512\\1\times1\times2048\end{pmatrix}$	$3\times\begin{pmatrix}1\times1,512\\3\times3\times512\\1\times1\times2048\end{pmatrix}$	$3\times\begin{pmatrix}1\times1,512\\3\times3\times512\\1\times1\times2048\end{pmatrix}$
	1x1	Avgpool-FC1000-Softmax				
Flops		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figura 8. Componentes de cada una de las arquitecturas Resnet

3.4.3 Densenet-201

Densenet [21] es la red más grande que ha sido utilizada, esta contiene 709 capas de las cuales se extrae información en 201 de ellas. La principal diferencia con Resnet es que esta tiene cuenta en cada capa todas las anteriores y envía todas a las posteriores. A diferencia de Resnet que realiza una adición en cada nivel, Densenet los concatena todos. Esto consigue una gran mejora en la eficiencia. En una arquitectura tradicional hay L conexiones distintas mientras que en Densenet hay L*(L+1)/2 conexiones distintas. La siguiente **figura [22]** muestra un de la arquitectura Densenet en la cual se puede apreciar cómo se envía la información a capas posteriores.

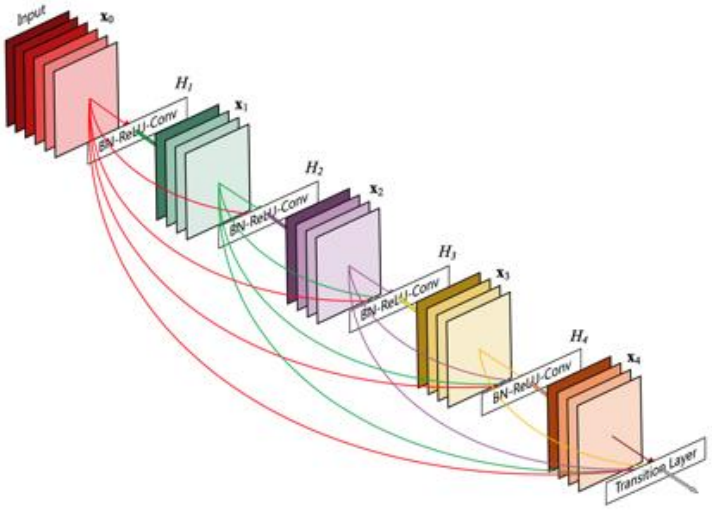

Figura 9. Arquitectura de una red convolucional Densenet

3.4.4 VGG-16

Esta red propuesta por el autor del artículo [23], abreviatura de *Visual Geometry Group*, está formada por 16 capas con pesos de las cuales 13 son capas convolucionales con filtros de tamaño 3x3, 2 capas fully-connected (*fc*) y una capa *Softmax*. La configuración de las capas fc sigue la misma estructura que la vista anteriormente en Alexnet. Las capas convolucionales están agrupadas en 5 grupos, los dos primeros son grupos de dos capas convolucionales y los 3 últimos tienen 3. Esta red muestra que la profundidad de estas es un parámetro importante pero que se paga con la gran cantidad de parámetros de entrenamiento. La **Figura 10 [24],** muestra un esquema de su arquitectura.

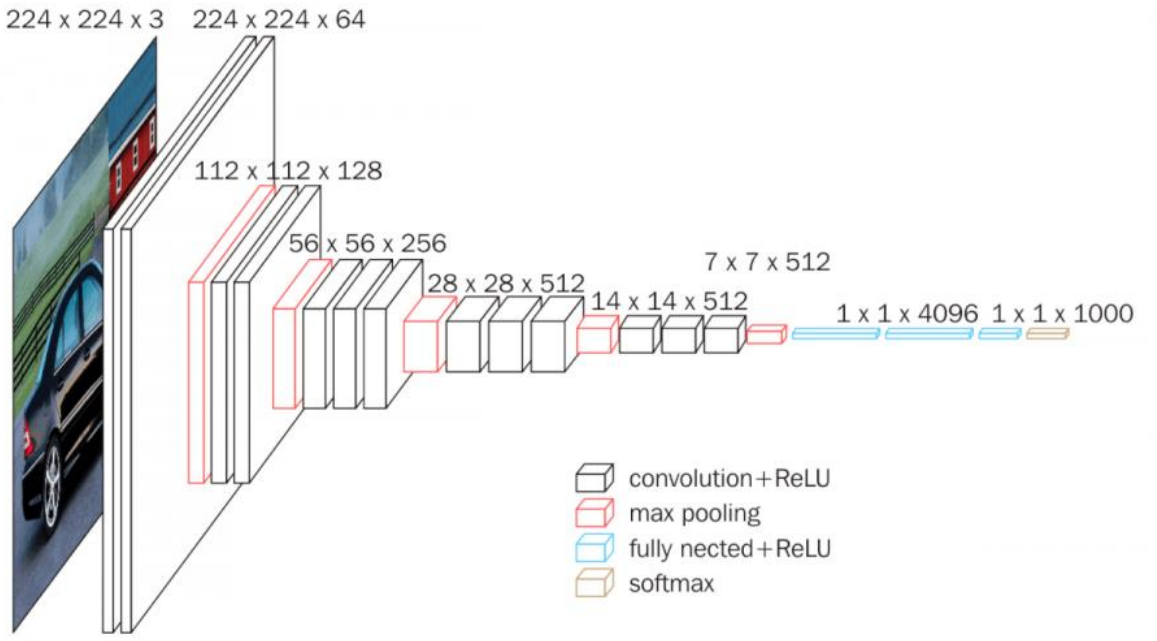

Figura 10. Arquitectura de la red VGG-16

Una vez vistas las arquitecturas de las redes, vamos a ver como se entrena una red neuronal y como se han entrenado durante el desarrollo del TFG las redes pre-entrenadas Resnet-18 y VGG-16.

3.5 Entrenamiento de una red neuronal

El entrenamiento es el proceso mediante el cual aprende una red neuronal, el objetivo principal de este es que la función de ajuste (porcentaje de clases que la red predice de manera correcta) se aproxime al máximo y que la función de pérdidas sea lo más mínima posible. En el entrenamiento de una red entran en juego múltiples parámetros, como por ejemplo el peso (*weight*) y el valor de sesgo (*bias*). En el entrenamiento se diferenciará entre dos tipos de datos de entrada, la de entrenamiento o *train* y las de validación o *test*.

En la figura vemos un esquema de las fases de un proceso de re-entrenamiento, este se divide en 4 pasos:

- Cargar la red pre-entrenada: En este paso se cargan los datos de la red que queremos re-entrenar.
- Sustitución de capas: Aquí modificaremos las capas con el fin de ajustar los pesos y el *bias* en las capas de las que se aprende y que se adapten a las nuevas clases que vamos a introducir.
- Entrenamiento de la red: Aquí es donde se hace que la red aprenda, se introducirán las nuevas clases y se le configurarán unas opciones de entrenamiento para conseguir que el aprendizaje sea lo más eficaz posible.
- Predicción y evaluación del ajuste: Se calculará el ajuste en la clasificación a la hora de predecir los datos. Para ver la eficacia de nuestro modelo se comprobarán nuestros datos de entrenamiento con las imágenes de *test*.

Una vez se ha entrenado la red podremos ver nuestros resultados de ajuste.

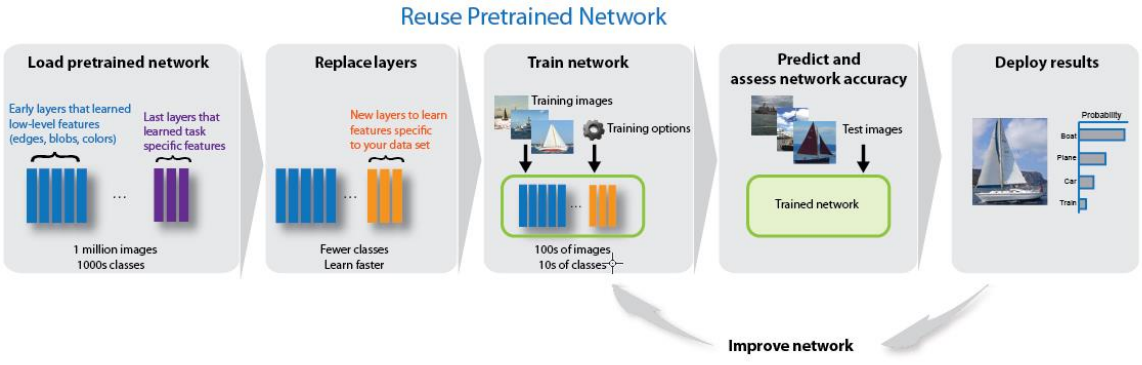

Figura 11. Esquema del proceso de aprendizaje de una red neuronal

4 Integración

4.1 Introducción

En este capítulo se mostrarán los datasets utilizados para la evaluación y el entrenamiento y veremos cómo implementar todas las técnicas de extracción de características, así como los parámetros a la hora de entrenar las redes todo ello visto en el **Capítulo 3**. Después haremos un análisis de las pruebas realizadas y de los resultados obtenidos para así poder valorar la eficacia de nuestros métodos de extracción de características en el ámbito de la re-identificación de personas.

4.2 Datasets

Tanto las imágenes procedentes de la fase de detección como las procedentes de la base de datos, serán de varios datasets, utilizando dos para entrenarlos con las redes citadas anteriormente y evaluándolos sobre el otro para ver los resultados y después analizar un tercero para ver cual dataset y que red obtienen el mejor resultado. Para esto utilizaremos los datasets Market1501 y DukeMTMCReID para el entrenamiento y la evaluación y Viper para evaluar y comparar los otros dos datasets. Estos presentan la siguiente estructura:

4.2.1 DukeMTMC4ReID

Este propuesto por el autor del artículo [15] dataset contiene 46261 imágenes de 1852 personas tomadas por 8 cámaras. De todas estas imágenes, 21551 son falsos positivos y 439 son distractores. A diferencia de los otros dos datasets utilizados, las imágenes tienen tamaños que varían entre los 72x34 píxeles y los 415x188 píxeles Como se puede apreciar en la **Figura 12 [15]**, se produce solapamientos solo entre los pares de cámaras 2-8 y 3-5.

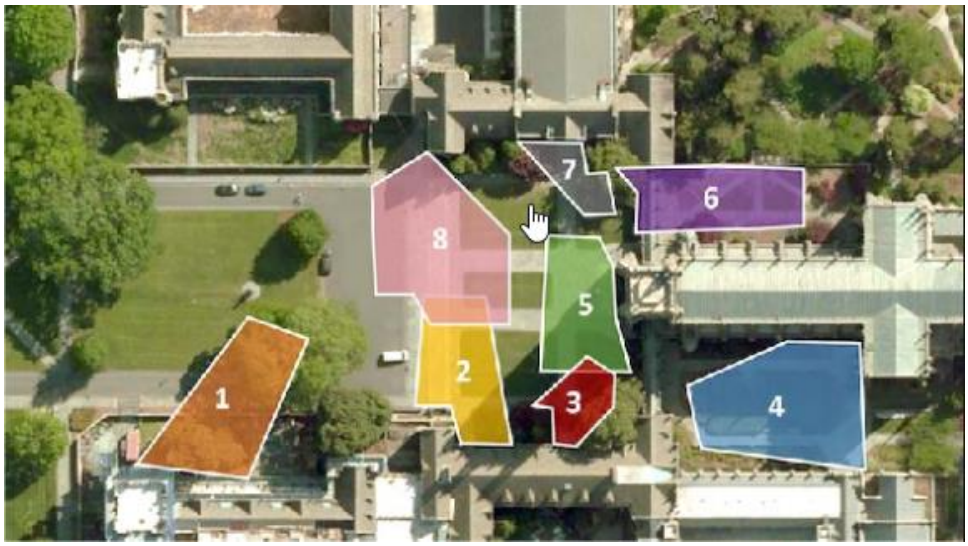

Figura 12. Disposición de las cámaras de DukeMTMC4ReID

4.2.1 Market1501

Formado por 32217 imágenes de 1501 personas tomadas por 6 cámaras. Estas personas aparecen al menos en dos cámaras distintas. En la **Figura 13 [16]** podemos ver algunas imágenes del dataset y como todas están redimensionadas a un tamaño de 128x64.

Figura 13. Imágenes de ejemplo del dataset Market-1501

4.2.2 ViPER

Este es el menos complejo de los 3 que se han probado en este TFG, compuesto por 1264 imágenes de 632 personas en 2 cámaras haciendo un total de 1264 imágenes (dos por cada persona). La **Figura 14 [17]** nos muestra un ejemplo de las imágenes del datasets, todas están redimensionadas a un tamaño de 128x48.

Figura 14 Ejemplo de imagenes del dataset ViPER

4.3 Integración

A continuación, se mostrarán los ajustes de la configuración para entrenar cada red neuronal y las nuevas partes implementadas para la utilización de estas para la extracción de características en el código del benchmark.

4.3.1 Entrenamiento de redes neuronales

Esta función sigue con la línea de lo explicado en la **Sección 3.5**, ahora veremos los parámetros específicos a fijar en la configuración de cada red para ser re-entrenada. Aquí solo se hablará de las redes que se han entrenado en este TFG.
Uno de los parámetros a tener en cuenta son las capas que se congelan de la red, estas son capas que no se re-entrenarán, son las encargadas de aprender características a bajo nivel semántico, genéricas para cualquier objeto y ya pre-entrenadas y por tanto son congeladas al no necesitar modificación. En el caso de Alexnet y de VGG-16 se congelarán las 5 primeras y para Resnet-18 se congelarán las 36 primeras capas. Esto se hace llamando a la función freezeWeights.m.

4.3.2 Implementación de las nuevas características

Para evaluar en nuestro entorno de benchmark, ha habido que realizar algunas modificaciones en algunos scripts. Este trabajo está enfocado a las medidas de extracción de características por tanto veremos cómo introducir nuevos métodos para poder evaluarlos.
El script principal, run_experiment_benchmark.m sirve para seleccionar los parámetros que podemos definir de cara a la evaluación, en este se pueden cambiar los parámetros relacionados con:

- Extracción de características (*Feature Options, fopts*)
- Métricas de aprendizaje (*Metric Options, mopts*)
- Evaluación (*Ranking Options, ropts*)

En el caso de interés, los parámetros de extracción de características nos dejarán elegir parámetros como pca o que característica queremos extraer. Una vez se ejecuta el script llamará a run_one_experiment.m y aquí se le pasarán los parámetros fopts y las imágenes a la función computeFeatures.m que será la encargada de extraer las características.

4.3.2.1 ComputeFeatures

Aquí se han introducido las nuevas características a evaluar, esta función tiene un apartado para cada método. Para introducir uno nuevo bastará con abrir una nueva opción para cuando se le llame y dentro de ella escribir el código de nuestra medida o bien llamar a la función o funciones encargadas de extraerla. Un ejemplo de esto para una característica tradicional es el caso de 'meancolor' la cual fue introducida y calcula la media de las componentes de cada canal de las imágenes. Para el caso de las características basadas en redes neuronales esto funcionará de una manera similar. Para extraer las características se introducirá la red que se quiere usar, la función de activación junto con los parámetros de entrada que esta necesita, entre ellos estará la capa de la que extraer la información la cual variará según la red, esto se explicará en la siguiente sección.
La función devolverá un vector de características y las opciones de la característica utilizada.

4.3.2.2 Capas de extracción de características

De estas capas será de las que se saque la información calculada por la red neuronal. Debido a las distintas arquitecturas, esta capa será distinta en función de la red utilizada y será escogida por que es la que mejor resultados da. Para Alexnet y VGG-16 se utilizará la capa fc7, para Resnet-18 y Resnet-101 se extraerán los datos de pool5 y para densenet201 de avg_pool.

4.3.2.3 Herramientas utilizadas

Para la realización de este TFG se ha utilizado la herramienta MATLAB r2018b y dentro de este una herramienta para el diseño e implementación de redes neuronales llamada Deep Learning Toolbox. Esta herramienta nos ha permitido utilizar redes neuronales pre-entrenadas, editar sus arquitecturas, entrenarlas y visualizar todo este proceso para supervisar el entrenamiento.

5 Pruebas y resultados

5.1 Introducción

En este apartado se mostrarán primero los resultados obtenidos tras entrenar las redes con los dataset de entrenamiento utilizados (DuleMTMC4ReID y Market1501) mediante gráficas que muestran el porcentaje de ajuste y de pérdidas para las redes Alexnet, Resnet-18 y VGG-16. En todos ellos se han utilizado las mismas opciones de entrenamiento, a excepción de las capas congeladas y de las épocas utilizadas con el fin de hacer una comparación basada en unas condiciones lo más similares posibles. Después se procederá a ver los resultados de todas las medidas de extracción de características utilizadas en cada dataset y realizar una comparativa. Tanto para el entrenamiento como para la evaluación, los datasets han de tener una estructura determinada. Para entrenar la red, se creará un directorio para cada persona en el que se introducirán todas sus imágenes, por eso este tipo de aprendizaje es supervisado como ya se comentó en la **Sección 3.3.** Para ejecutar el benchmark, este necesita que haya un directorio por cada cámara y dentro un directorio de cada persona.

5.2 Entrenamiento en DukeMTMC4ReID

A la hora de entrenar con el dataset de DukeMTMC4ReID, hay algunos aspectos a tener en cuenta de los comentados anteriormente en la **Sección 4.2.1**, como que las imágenes tienen tamaños distintos o la omisión de los distractores para realizar el entrenamiento. Hemos obtenido diferencias notables en las 3 gráficas de re-entrenamiento que vamos a ver a continuación:

5.2.1 Alexnet

En cuanto al ajuste (*Accuracy*) y las pérdidas (*Loss*), los resultados de validación de ajuste de Alexnet son bastante bajos (27,1%) y la gráfica de pérdidas tampoco presenta resultados buenos. Se puede apreciar por el aplanamiento de la curva que aumentando el número de épocas no hubiera presentado mejores resultados. Este bajo ajuste se puede deber a que esta red presenta una arquitectura de poca complejidad y entonces no consigue unos buenos resultados para un modelo de este tamaño o por las diferencias de las dimensiones entre imágenes. La **Figura 15.** muestra las gráficas de *Accuracy* y *Loss* resultantes del re-entrenamiento de Alexnet.

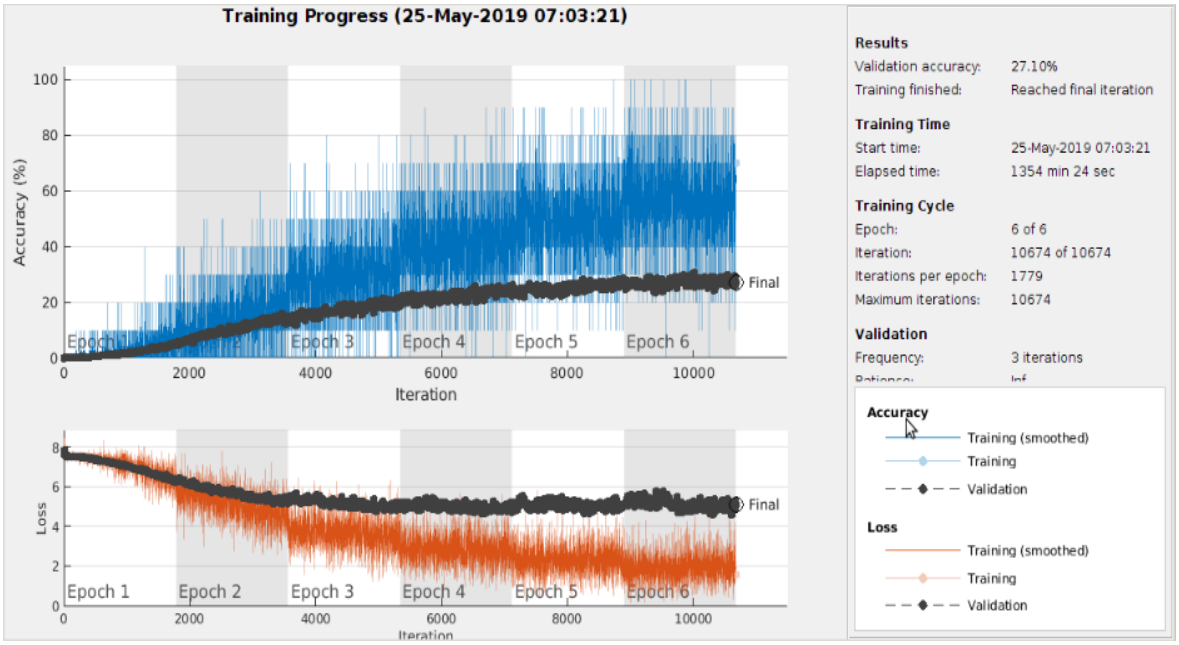

Figura 15. Gráfica de entrenamiento del dataset DukeMTMC4ReID en la red Alexnet

5.2.2 Resnet-18

Respecto a Resnet-18, se aprecia una mejora en ambas funciones, especialmente en los datos de entrenamiento de las dos, esto significa que la red ha entrenado bien. En cuanto a la validación, el porcentaje de *Accuracy* (43,25%) ha mejorado respecto de Alexnet pero sigue sin ser alto, tiene una curva de comportamiento muy similar a la de entrenamiento pero notablemente más baja. A la función de pérdidas le ocurre exactamente lo mismo que a la de ajuste, la gráfica de Loss muestra una gran disminución para los datos de entrenamiento con una curva similar a la de validación siendo esta segunda más moderada. Esta diferencia puede ser debida a que el modelo entrena bien los datos, pero a la hora de probar con los de test no consigue validarlos con los de entrenamiento. En la **Figura 16.** Se pueden ver las gráficas de comportamiento de *Accuracy* y *Loss* tras re-entrenar Resnet-18.

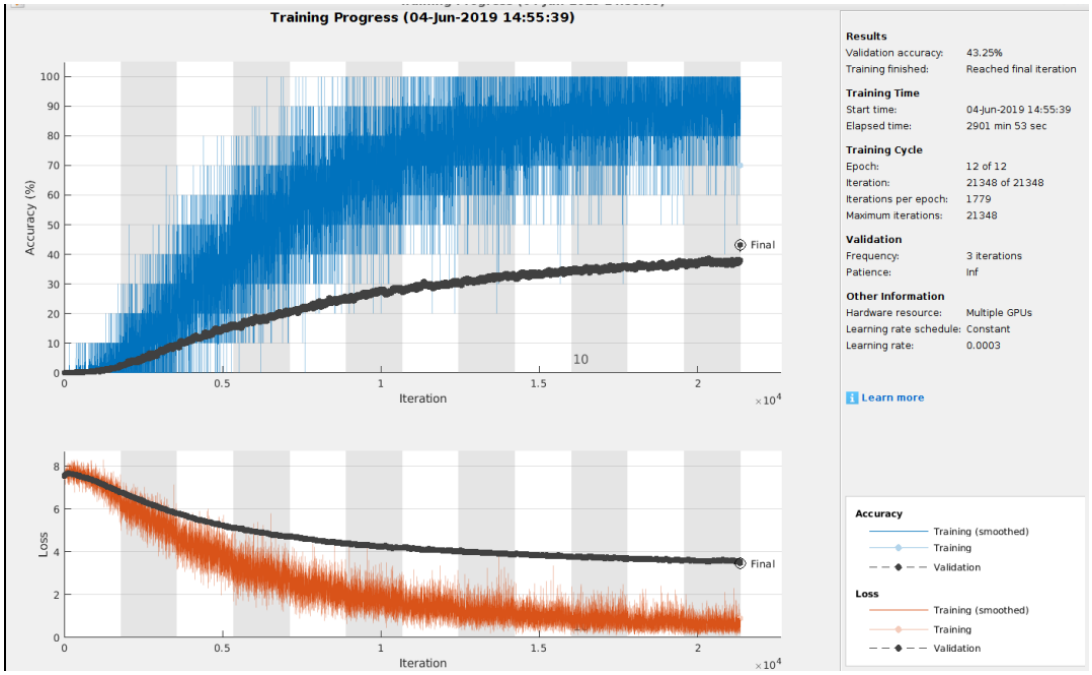

Figura 16. Gráfica de entrenamiento del dataset DukeMTMC4ReID en la red Resnet-18

5.2.1 VGG-16

Para el caso de VGG-16, el comportamiento es bastante parecido al Resnet-18, tanto para Accuracy como para Loss hay buenos porcentajes en cuanto a los datos de entrenamiento. Los datos de validación mantendrán una línea similar, pero algo más moderados. Un aspecto que salta a la vista al comparar la gráfica de Resnet-18 y la de VGG-16 es que los datos de validación en esta segunda se presentan algo más dispersos esto se puede deber a que VGG-16 tiene una cantidad mucho más alta de parámetros a modelar que Resnet-18 haciendo más difícil el ajuste en los datos de test. En la **Figura 17.** se pueden visualizar los datos que acaban de ser comentados.

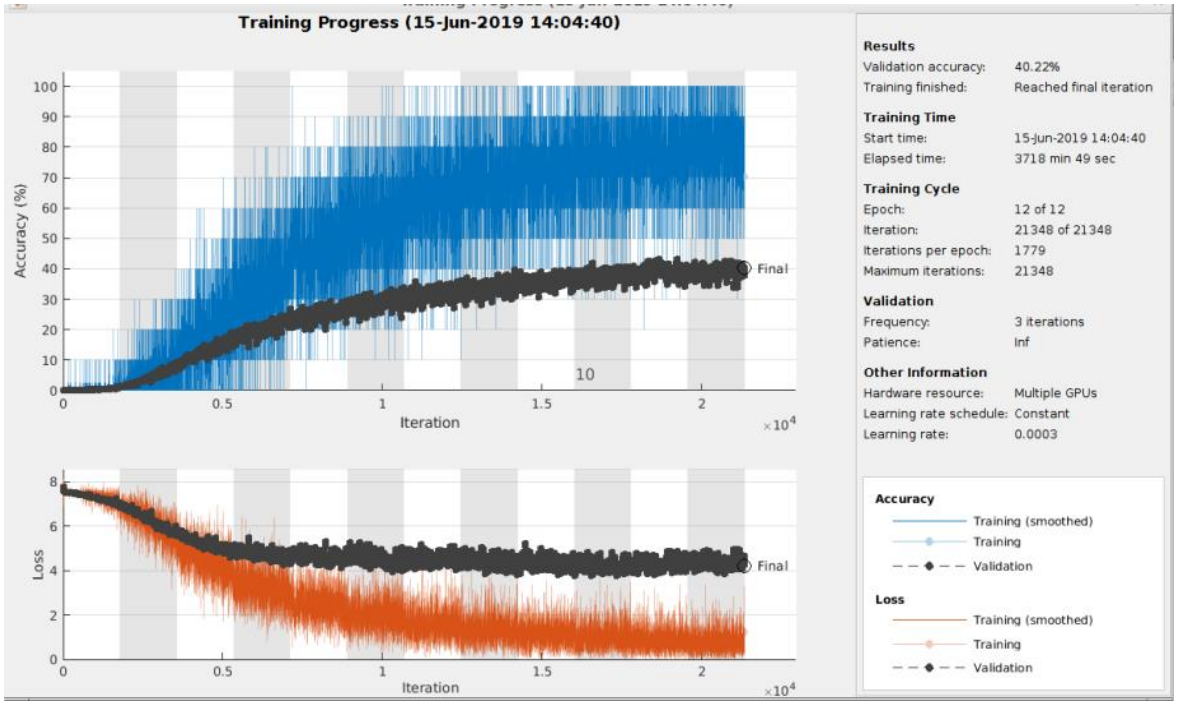

Figura 17. Gráfica de entrenamiento del dataset DukeMTMC4ReID en la red VGG-16

5.3 Entrenamiento en Market1501

En este caso y como ya comentamos en la **sección 4.2.2** el dataset de Market1501 presenta todas las imágenes con el mismo tamaño, para esta tarea se han extraído los distractores como en los casos anteriores. Este dataset se ha utilizado para entrenar las mismas redes pre-entrenadas que el dataset citado en la sección anterior, Alexnet, Resnet-18 y VGG-16. A continuación procederemos a ver cada uno de los resultados con este dataset.

5.3.1 Alexnet

Esta gráfica tiene unos resultados que no son tan bajos como en su homólogo de la sección anterior pero tampoco llegan a cifras altas. Tanto los valores de test como los de validación tienen unos resultados que no distan demasiado unos de los otros, es decir, tanto el entrenamiento como la evaluación tienen una tasa de mejora muy parecida. Estos resultados los podemos ver en la **Figura 18.**

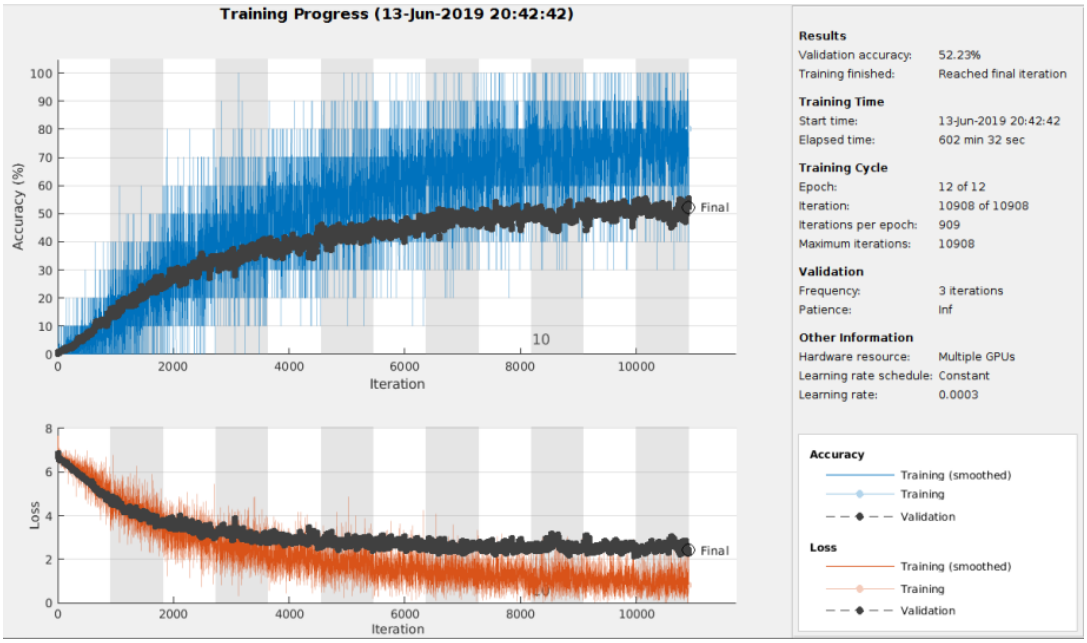

Figura 18. Gráfica de entrenamiento del dataset Market1501 en la red Alexnet

5.3.2 Resnet-18

Esta gráfica presenta unos resultados levemente mejores a Alexnet, alcanzando un 63,12% de ajuste en la validación, la distancia entre los valores de entrenamiento y los de test es parecida siendo efectiva tanto en entrenamiento como en evaluación. En la **Figura 19.** podremos visualizar estos resultados.

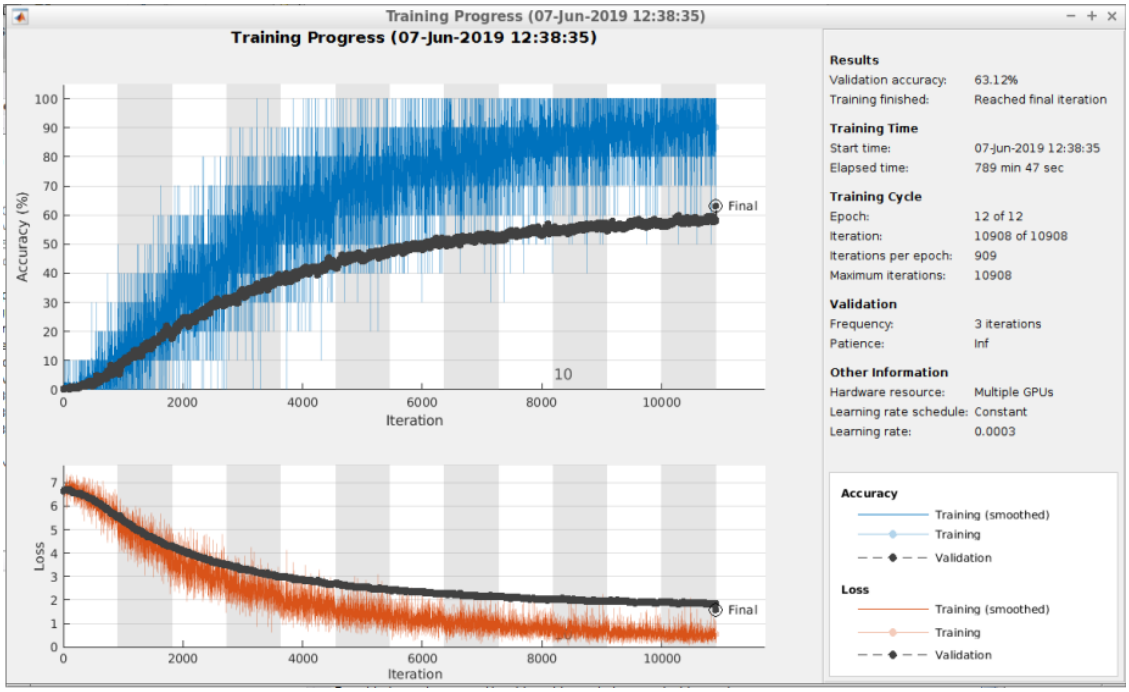

Figura 19. Gráfica de entrenamiento del dataset Market1501 en la red Resnet-18

5.3.3 VGG-16

Esta última presenta un resultado muy similar a la anterior, distando su valor final (58,53%) muy poco del de Resnet-18. Los datos de entrenamiento y los de test se comportan de manera similar y su diferencia más notable es la variación entre los valores de test esto se puede observar en la **Figura 20**.

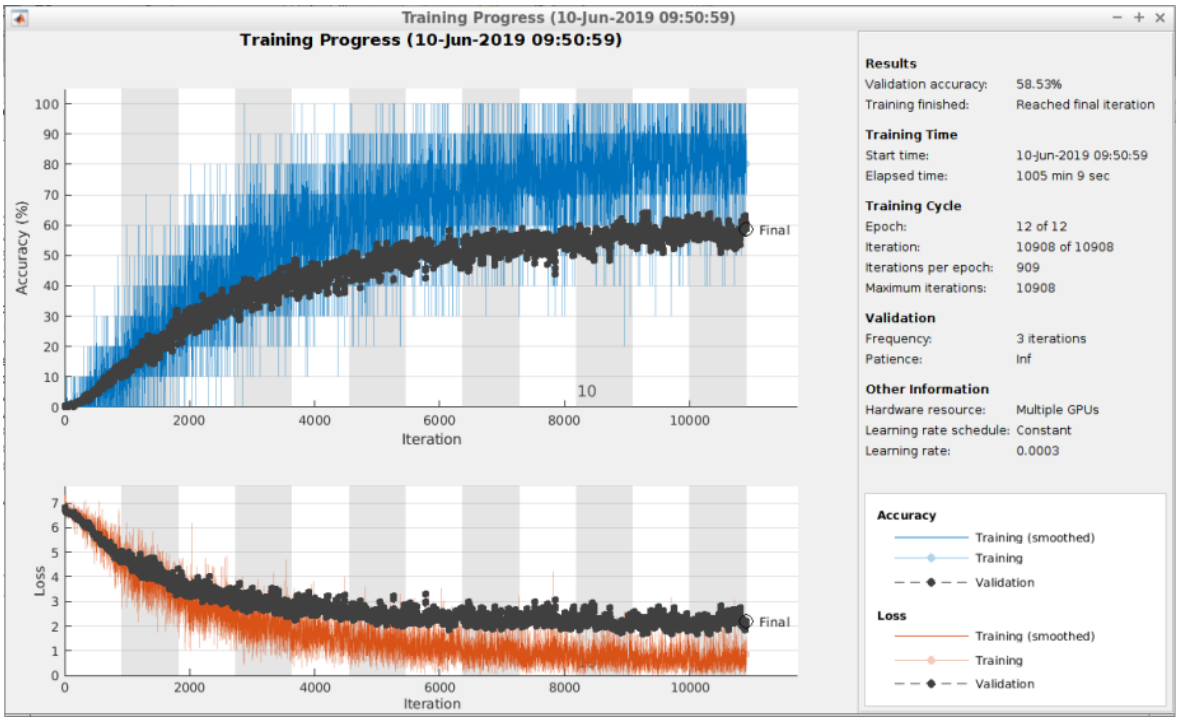

Figura 20. Gráfica de entrenamiento del dataset Market1501 en la red VGG-16

5.4 Conclusiones de las gráficas

Una vez vistas todas las gráficas podemos observar que las redes entrenadas con el dataset DukeMTMC4ReID son peores que los obtenidos en Market, esto se puede deber a los cambios de tamaño entre las distintas imágenes.

5.5 Resultados de entrenamiento

Por el contrario, para el caso del tiempo es notablemente más pequeño en Market1501 esto se debe a que el dataset es bastante más pequeño (entorno a un 25%) y por tanto habrá menos imágenes que procesar.
Si enfocamos los resultados por red, la que mejores resultados ha presentado en ambos datasets ha sido Resnet-18 sin ser la que más tiempo necesita para realizar el entrenamiento. En la **Tabla 1.** se pueden observar los resultados obtenidos ordenados en una tabla cuyas columnas se corresponden con el porcentaje de ajuste del dataset utilizado seguido del tiempo de entrenamiento.

Red	DukeMTMC4ReID	Tiempo	Market1501	Tiempo
Alexnet	27,1	1354 minutos	52,23	602 minutos
Resnet-18	43,25	2901 minutos	63,12	789 minutos
VGG-16	40,22	3718 minutos	58,53	1005 minutos

Tabla 1. Resultados de entrenamiento y tiempos de ejecución

Tras haber entrenado las redes neuronales, ya disponemos de todos los métodos de extracción de características, tanto los tradicionales como los basados en redes neuronales. En el siguiente apartado se evaluarán todos los métodos en los 3 datasets escogidos para las pruebas.

5.6 Evaluación en DukeMTMC4ReID

Al evaluar este dataset, en el apartado de los métodos de extracción de características manuales, se puede observar que el método que presenta peores resultados es Meancolor (Color medio, prueba en computeFeatures.m para nuevas integraciones) con diferencia, este es demasiado simple y por tanto carece de utilidad. El mejor es WHOS (*Weighted Histogram of Overlapping Stripes*) las diferencias entre el mejor resultado y el peor en cada uno los 4 Ranks evaluados es considerable. Para el caso de los métodos basados en redes neuronales, el mejor resultado obtenido ha sido Resnet-18 tras ser re-entrenada con Market1501 distando mucho de los resultados de la red pre-entrenada VGG-16. Finalmente, el mejor resultado para este dataset ha sido WHOS para Rank1 y Rank20 y Resnet-18 re-entrenada para Rank5 y Rank10. La **Tabla 2.** Muestra todos los resultados de los distintos métodos evaluados en DukeMTMC4ReID.

MAN/CNN	TYPE	RANK1	RANK5	RANK10	RANK20
MANUAL	WHOS	28,03	44,37	53,3	62,61
	gBiCov	10,63	23,04	31,71	41,61
	MEANCOLOR	1,23	4,69	7,33	11,62
	LDFV	24,43	41,55	49,1	58,42
	COLOR_TEXTURE	17,36	31,63	41,09	50,35
	HIST_LBP	12,98	26,49	34,02	43,13
CNN	RESNET101	19,05	34,89	45,09	53,49
	DENSENET201	19,74	34,39	43,08	51,29
	Alexnet	17,32	32,09	38,59	46,85
	Alexnet_MARKET	22,16	39,24	47,22	56,83
	RESNET18	11,4	23,97	31,19	40,1
	RESNET18_MARKET	25,42	43,87	57,9	60,5
	VGG16	8,2	20,27	26,43	35,39
	VGG16 MARKET	25,56	40,31	47,85	55,73

Tabla 2. Resultados obtenidos sobre el dataset DukeMTMC4ReID

5.7 Evaluación en Market1501

Descartando los resultados obtenidos con el método Meancolor, el mejor resultado de los métodos tradicionales es WHOS mientras que el peor es gBiCov teniendo un rendimiento bastante inferior. Para el caso de los métodos basados en redes neuronales, el mejor método ha sido Resnet-18 tras ser re-entrenada y el peor VGG-16 con un rendimiento muy bajo en comparación con todos los demás. El mejor método en este dataset ha sido WHOS, teniendo mejores resultados en los 4 Ranks. La **Tabla 3.** Muestra todos los resultados de los distintos métodos para el dataset Market1501.

MAN/CNN	TYPE	RANK1	RANK5	RANK10	RANK20
MANUAL	**WHOS**	**40,02**	**63,93**	**73,49**	**81,35**
	gBiCov	19,03	38,21	48,69	59,41
	MEANCOLOR	1,01	3,77	6,5	11,52
	LDFV	31,26	55,31	66,33	76,13
	COLOR_TEXTURE	28,03	49,91	60,69	70,4
	HIST_LBP	24,35	45,64	55,85	66,24
CNN	**RESNET101**	17,01	37,35	48,25	59,09
	DENSENET201	16,21	37,17	46,73	57,84
	Alexnet	19,66	39,9	49,58	60,57
	Alexnet_DUKE	23,99	43,17	51,9	61,61
	RESNET18	10,21	24,11	33,31	43,68
	RESNET18_DUKE	36,46	59,29	68,71	77,38
	VGG16	7,48	19,09	27,26	37,5
	VGG16 DUKE	30.55	52.67	62.86	72.18

Tabla 3. Resultados obtenidos sobre el dataset Market1501

5.8 Evaluación en ViPER

En este dataset, comprobaremos los dos re-entrenamientos de cada red, es decir, habiendo re-entrenado las redes con DukeMTMC4ReID y con Market1501.
Para el caso de los métodos manuales, el mejor resultado ha sido LDFV y el peor gBiCov, estando bastante alejado de los otros métodos. En el caso de los métodos no tradicionales, el mejor ha resultado ser Resnet-18 re-entrenado con el dataset DukeMTMC4ReID y el peor VGG-16 con resultados muy inferiores a todos menos a Resnet-18. El método que mejores resultados ha tenido para todos los Ranks ha sido LDFV, aunque WHOS ha tenido unos resultados muy parecidos. La **Tabla 4.** muestra todos los resultados de los distintos métodos en el dataset ViPER.

MAN/CNN	TYPE	RANK1	RANK5	RANK10	RANK20
MANUAL	**WHOS**	24,92	53,98	68,39	82,28
	gBiCov	9,91	24,59	34,19	47,1
	MEANCOLOR	1,66	6,53	12,88	23,28
	LDFV	**27,07**	**56,16**	**70,4**	**83,8**
	COLOR_TEXTURE	22,15	51,16	69,97	78,27
	HIST_LBP	20,62	46,19	61,17	75,89
CNN	**RESNET101**	14,56	36,16	49,18	64,46
	DENSENET201	12,06	32,33	44,19	59,62
	Alexnet	11,41	29,76	41,79	56,17
	Alexnet_DUKE	11,79	28,13	38,35	51,09
	Alexnet_MARKET	18,73	39,72	51,34	65,06
	RESNET18	6,2	19,22	28,16	42,23
	RESNET18_DUKE	22,12	45	58,18	72,45
	RESNET18_MARKET	18,78	41,47	53,56	67,34
	VGG16	4,35	15,27	24,53	37,64
	VGG16 DUKE	17.07	36.46	46.87	58.5
	VGG16 MARKET	18,08	34,1	44,26	57,53

Tabla 4. Resultados obtenidos sobre el dataset ViPER

5.9 Evaluación global

Tras realizar las pruebas en los 3 datasets, hemos visto que los mejores resultados por dataset han sido con:

- DukeMTMC4ReID: WHOS (Tradicional)
- Market1501: WHOS (Tradicional)
- ViPER: LDFV(Tradicional)

El mejor resultado en la evaluación global ha sido LDFV en el dataset de ViPER además de tener los resultados medios más altos, por el contrario, los resultados medios más bajos han sido en el dataset de DukeMTMC.
En ninguno de los casos hemos obtenido el mejor resultado con una red neuronal, pero sí que ha habido alguno que se ha aproximado o ha logrado buenos resultados. En la Figura se pueden ver de manera gráfica los resultados obtenidos.

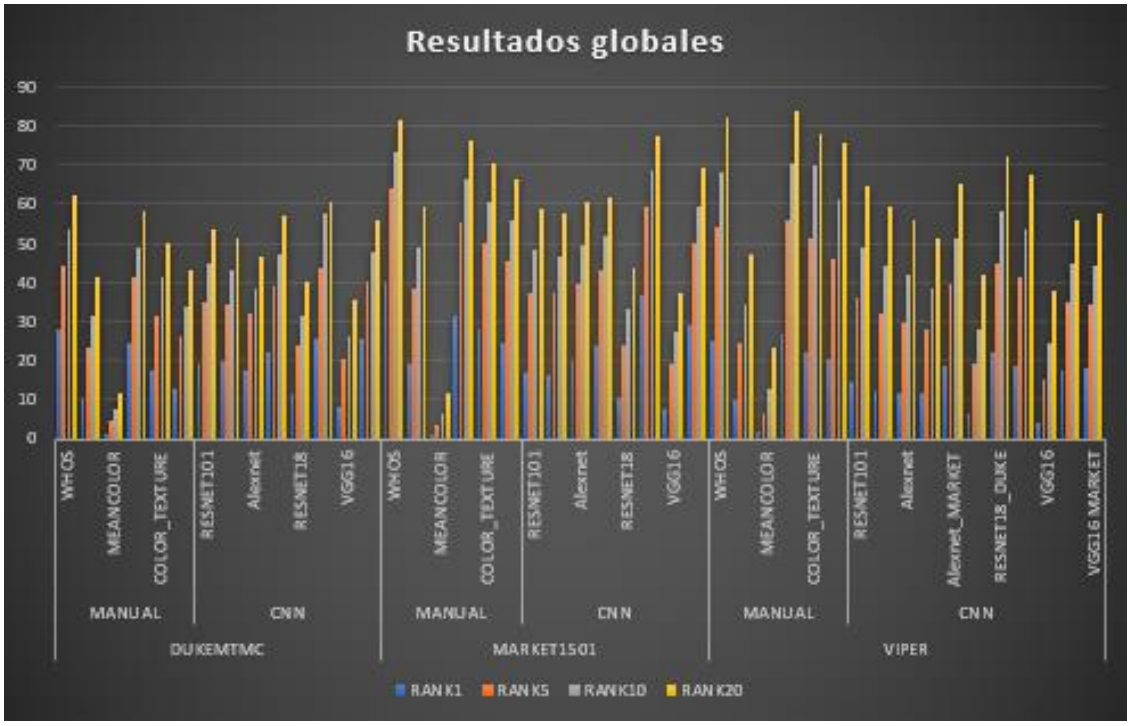

Figura 21. Gráfica global de los resultados obtenidos

6 Conclusiones y trabajo futuro

6.1 Conclusiones

El objetivo principal de este TFG es evaluar y comparar los resultados obtenidos de evaluar los métodos de extracción de características manuales y los basados en redes neuronales, todas estas pruebas en unas condiciones y con unos parámetros iguales y sobre los mismos datasets con el fin de crear un entorno común de evaluación.
Para conseguir este objetivo, se ha estudiado el estado del arte en el **Capítulo 2**, apoyándose en los artículos [1][2][3][13][14] , apoyando la utilización de redes neuronales y en concreto el aprendizaje profundo junto con diversas técnicas como la segmentación de las personas en varias partes para una mejor identificación.
En el **Capítulo 3**, se han establecido todos los métodos de extracción de características utilizados como WHOS, GOG, gBiCov, LDFV, Color & Texture, Histogram LBP y LOMO para el caso de los métodos tradicionales y la utilización de las redes pre-entrenadas Alexnet Densenet-201 Resnet y VGG-16 para los métodos basados en redes neuronales.
En el **Capítulo 4** se muestran los distintos datasets utilizados y sus características además de la implementación de nuevas características y las herramientas utilizadas para realizar este TFG.
En el **Capítulo 5** se muestran los resultados obtenidos tras el re-entrenamiento de las redes Alexnet, Resnet-18 y VGG-16, una comparativa de los resultados dividida en por los datasets utilizados y una comparativa global de los resultados en forma de gráfica. Estas comparativas nos dejan ver que los resultados obtenidos han sido mejores para los métodos tradicionales, en concreto para 2 de ellas (WHOS y LDFV) pero que las redes neuronales pueden obtener buenos resultados, sobre todo Alexnet y Resnet-18 tras ser re-entrenadas con los datasets y que estas son mucho más rápidas de ejecutar una vez han aprendido. Esta mejoría se puede apreciar en las tablas ya que aparecen las redes antes y después de ser re-entrenadas, la mejoría es notable en todas y como estas puede mejorar aún más introduciendo más datos.

6.2 Trabajo futuro

Tras ver los resultados obtenidos en este TFG, se plantea trabajar con redes neuronales más nuevas y complejas utilizando mayor número de imágenes y de personas para el entrenamiento y evaluar utilizando métricas de entrenamiento distintas. Es importante continuar observando cómo se comportan las redes neuronales al ser entrenadas con imágenes en condiciones distintas y cómo se comportan al utilizar las redes en condiciones de el mismo tipo.
En cuanto a los métodos tradicionales, se propone utilizar tanto los métodos basados en el estado del arte que presenten mejores resultados para los datasets utilizados a la hora de realizar las comparaciones futuras con los métodos basados en redes neuronales sirviendo así de referencia.

En cuanto a los dataset, es recomendable realizar pruebas con otros distintos con condiciones diferentes, como variaciones del entorno o escenarios con condiciones adversas como oclusiones, baja iluminación o grandes grandes variaciones del fondo o como en los métodos de obtención de la imagen ya sean manuales o mediante métodos de detección de personas.

Referencias

[1] Martín-Nieto, R., García-Martín, Á., Martínez, J., & SanMiguel, J. (2018). "Enhancing Multi-Camera People Detection by Online Automatic Parametrization Using Detection Transfer and Self-Correlation Maximization", Sensors, 18(12), 4385.

[2] Yao, H., Zhang, S., Hong, R., Zhang, Y., Xu, C., & Tian, Q. (2019). "Deep Representation Learning With Part Loss for Person Re-Identification", In IEEE Transactions On Image Processing, 28(6), 2860-2871.

[3] Huang, Y., Xu, J., Wu, Q., Zheng, Z., Zhang, Z., & Zhang, J. (2019). "Multi-Pseudo Regularized Label for Generated Data", In Person Re-Identification. IEEE Transactions On Image Processing, 28(3), 1391-1403.

[4] Lisanti, G., Masi, I., Bagdanov, A., & Bimbo, A. (2015). "Person Re-Identification by Iterative Re-Weighted Sparse Ranking", In IEEE Transactions On Pattern Analysis And Machine Intelligence, 37(8), 1629-1642.

[5] T. Matsukawa, T. Okabe, E. Suzuki and Y. Sato. (2016). "Hierarchical Gaussian Descriptor for Person Re-identification", In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 1363-1372.

[6] B. Ma, Y. Su, and F. Jurie, (2012). "Local descriptors encoded by fisher vectors for person re-identification", In ECCV Workshops.

[7] Fogel, I., & Sagi, D. (1989). "Gabor filters as texture discriminator". In Biological Cybernetics, 61(2).

[8] V. Takala and M. Pietikainen. (2007). "Multi-object tracking using color, texture and motion", In IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[9] LBP y ULBP – Local Binary Patterns y Uniform Local Binary Patterns. (2019). Retrieved from https://cesartroyasherdek.wordpress.com/2016/02/26/deteccion-de-objetos-vi/

[10] Liao, S., Hu, Y., Zhu, X., Li, S.Z. (June 2015). "Person re-identification by local maximal occurrence representation and metric Learning", In The IEEE Conference on Computer Vision and Pattern Recognition.

[11] LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning", In Nature, 521(7553), 436-444.

[12] L. Bazzani, M. Cristani, A. Perina, M. Farenzena and V. Murino. (2010).“Multiple -Shot Person Re-identification”, by HPE Signature. Pro.ceedings of the 2010 International Conference on Pattern Recognition. 1413–1416

[13] Bazzani, L., Cristani, M., & Murino, V. (2013). “Symmetry-driven accumulation of local features for human characterization and re-identification”, In Computer Vision And Image Understanding, 117(2), 130-144.

[14] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. (2010). “Person re-identification by symmetry-driven accumulation of local features”, In Proc. IEEE Intern. Conf. on Computer Vision and Pattern Recognition.

[15] M. Gou et al., (2017). “DukeMTMC4ReID: A large-scale multi-camera person re-identification dataset,” In CVPR Workshops.

[16] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. (2015). “Scalable person re-identification: A benchmark”, In ICCV.

[17] Bouma, H., Borsboom, S., den Hollander, R., Landsmeer, S., & Worring, M. (2012). “Re-identification of persons in multi-camera surveillance under varying viewpoints and illumination”, In Sensors, And Command, Control, Communications, And Intelligence (C3I) Technologies For Homeland Security And Homeland Defense XI.

[18] Karanam, S.; Gou, M.; Wu, Z.; Rates-Borras, A.; Camps, O.; and Radke, R. J. (2016). “A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets”.

[19] Convolutional Neural Network Models - Deep Learning. (2019). https://www.slideshare.net/mohamedloey/convolutional-neural-network-models-deep-learning

[20] Ruixin Zhang Qiuyu Zhu. Henet. (2018). “A highly efficient convolutional neural networks optimized for accuracy, speed and storage”.

[21] G. Huang, Z. Liu, K. Q. Weinberger, and L. Maaten. (2017). “Densely connected convolutional networks”, In CVPR.

[22] Common architectures in convolutional neural networks. (2019). https://www.jeremyjordan.me/convnet-architectures/

[23] K. Simonyan and A. Zisserman. (2015). “Very deep convolutional networks for large-scale image recognition”, In ICLR.

[24] VGG16 - Convolutional Network for Classification and Detection. (2019). https://neurohive.io/en/popular-networks/vgg16/

[25] An Intuitive Explanation of Convolutional Neural Networks. (2016). https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[26] R. A. Fisher. (1936). “The use of multiple measurements in taxonomic problems”, In Annalsofeugenics (AE), vol.7, no.2, pp.179–188.

[27] S. Pedagadi et al. (2013). “Local fisher discriminant analysis for pedestrian re-identification,” In CVPR.

[28] F.Xiongetal. (2014). “Personre-identificationusingkernel-basedmetric learning methods”, In ECCV.

[29] S. Yan et al., (2007) “Graph embedding and extensions: a general framework for dimensionality reduction”, T-PAMI, vol. 29, no. 1, pp. 40–51.

[30] S. Liao et al., (2015). “Person re-identification by local maximal occurrence representation and metric learning”, In CVPR.

[31] A. Mignon and F. Jurie, (2012). “PCCA: A new approach for distance learning from sparse pairwise constraints,” In CVPR.

[32] L. Zhang, T. Xiang, and S. Gong. (2016) “Learning a discriminative null spaceforpersonre-identification” inCVPR , pp.1239–1248.

[33] M. Koestinger et al. (2012) “Large scale metric learning from equivalence constraints,” in CVPR.

[34] Z. Li et al. (2013) “Learning locally-adaptive decision functions for person verification,” In CVPR.

[35] W.-S. Zheng, S. Gong, and T. Xiang. (2011) “Person re-identification by probabilistic relative distance comparison,” In CVPR.

Glosario

BIF	Biologically Inspired Features
CNN	Convolutional Neural Network
DPM	Deformable Parts Model
FC	Fully Connected
FDA	Fisher Discriminant Analysis
gBiCov	Biolgical Inspired features combined with Covatiance
KISSME	Keep-It-Simple-and-Straightforward-Metric
KLFDA	Kernelized Local Fisher Discriminant Analysis
KMFA	Kernelized Marginal Fisher Analysis
KPCCA	Kernelized Pairwise Constrained Component Analysis
LDFV	Local Descriptors encoded by Fiher Vector
LFDA	Local Fisher Discriminant Analysis
NFST	Null Foley-Sammon Transform
PCA	Principal Component Analysis
PCCA	Pairwise Constrained Component Analysis
PRDC	Probabilistic Relative Distance Comparison
ReLU	Rectified Linear Unit
SVMML	Support Vector Machine on Multi Layer
TFG	Trabajo de Fin de Grado
VGG	Visual Geometry Group
VPU	Video Processing and Understanding
WHOS	Weighted Histogram of Overlapping Stripes
XQDA	Cross-view Quadratic Discriminant Analysis